

# MTools 软件手册

2017



# WRO



蓝鲸智能机器人（深圳）有限公司



---

# Table of Contents

Introduction	1.1
1. Hardware Interface	1.2
2. Software Installation	1.3
3. Software Intro	1.4
4. Configure drive parameters	1.5
5. Test the motor phase and Hall wire wiring sequence	1.6
6. Drive firmware update	1.7
7. Calibrate the IMU	1.8
8. Used in ROS	1.9

- [introduction](#)

## introduction

This document introduces the use of the DR03 hub motor driver, including the connection of the driver, the control of the driver, the parameter setting of the driver, and the test of the driver.

[hub drive package download](#)

- [Drive Hardware Description](#)
  - [Overview](#)
    - [BW-DR03 drive parameter list](#)
    - [Features](#)
  - [Interface definition](#)
    - [Power input interface P2](#)
    - [Motor phase line interface Ma1, Mb1](#)
    - [Motor Hall line interface H0, H1](#)
    - [Infrared obstacle avoidance module interface C1, C4, C2, C3](#)
    - [Ultrasonic ranging module interface S1, S2](#)
    - [Serial communication interface P1](#)
  - [communication protocol](#)
    - [computer commands](#)
    - [Drive upload packets](#)

## Drive Hardware Description

### Overview

BW-DR03 is a high-performance, multi-function, low-cost brushless DC driver with Hall sensor developed by our company. The drive has its own speed closed-loop PID control, the speed control accuracy is within 1%, the low-speed output torque is large, and the high-speed braking is fast. It supports to control two motors at the same time, comes with IMU, and can be connected to 4 infrared obstacle avoidance modules and 2 ultrasonic ranging modules. The speed control and parameter configuration of the driver motor are realized through the rs232 serial port. With the configuration software of the host computer and the ROS driver package, a high-performance robot chassis can be quickly built.

### BW-DR03 drive parameter list

Parameter	Description
Motor drive tube input voltage	DC12V-46V
Motor Hall Type	120 degree
Working current	30A Max(15A Single)
Operating mode	Hall speed closed loop
Speed mode	RS232 serial port commands
IMU	MPU9250 with 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer
Infrared obstacle avoidance module	Support up to 4 channels
Ultrasonic ranging module	Maximum support 2 channels
working environment	Occasion: no flammable, explosive gas, no dust Temperature: -10-50 degrees Celsius Vibration: less than 0.5G, 10HZ-60HZ Protection level: not waterproof
Cooling method	natural air cooling
size	12815632
weight	500g

Features

Speed PID closed-loop control, large torque at low speed.

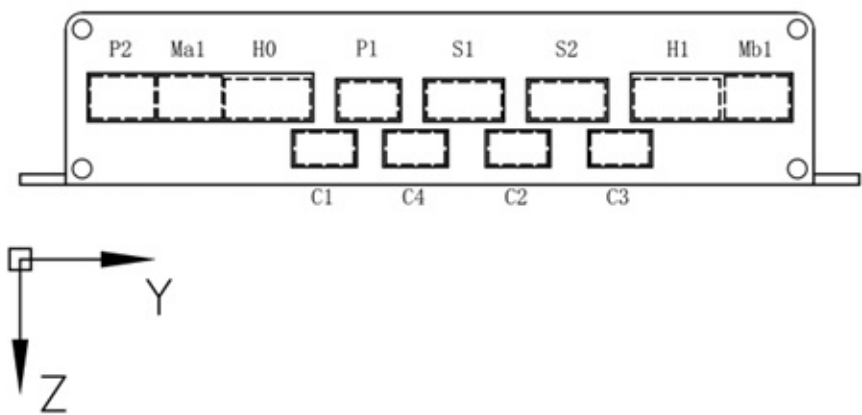
Wide speed range, 0-2000RPM.

Supports a wide range of motor voltages, 12V-36V.

It supports to control two motors at the same time, comes with IMU, and can be connected to 4 infrared obstacle avoidance modules and 2 ultrasonic ranging modules.

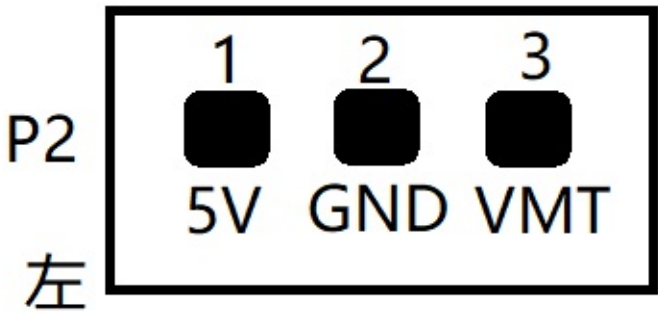
With ROS driver package, you can directly output topic data such as odometer and IMU.

Interface definition



Distribution diagram of the front interface of the drive

Power input interface P2



interface	description
1. 5V	RS232 version, this pin is abandoned, please do not connect anything
2. GND	power ground
3. VMT	Motor drive tube power positive input

The VMT voltage value is determined by the rated voltage of the motor. For example, a 36v motor can be connected to the positive pole of a 36v battery. Note that the battery must be used for direct power supply. Powering it through a dc regulated power supply will burn the driver.

## Motor phase line interface Ma1, Mb1



interface	description
1. mA	Motor phase wire U (yellow)
2. mb	Motor phase line V (green)
3. mC	Motor phase wire W (blue)

The definition of the color of the motor phase wire is related to the motor manufacturer. When actually wiring, please connect the motor phase wire to mA mB mC at will, and then adjust the Hall phase wire below (in theory, any motor phase wire connection has a set of corresponding Therefore, you can arbitrarily specify the wiring sequence of the motor phase wires). When the driver can normally control the forward and reverse rotation of the motor in the open-loop mode, it means that the wiring sequence of the Hall phase wires and the motor phase wires is correct. For details, please refer to the relevant chapters in the driver configuration software manual.

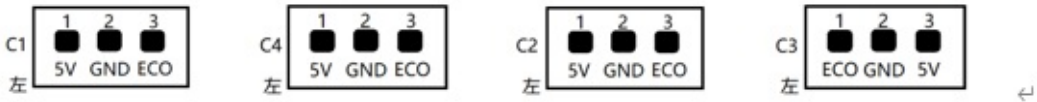
## Motor Hall line interface H0, H1



interface	description
1. 5V	Hall power supply positive
2. HA	Hall U line (yellow)
3. HB	Hall V line (green)
4. HC	Hall W line (blue)
5. GND	Hall power supply ground

H0 corresponds to Ma1 channel motor, H1 corresponds to Mb1 channel motor. When the driver is installed face down, Ma1 corresponds to the right wheel motor, and Mb1 corresponds to the left wheel motor; when the driver is installed face up, Ma1 corresponds to the left wheel motor, and Mb1 corresponds to the right wheel motor.

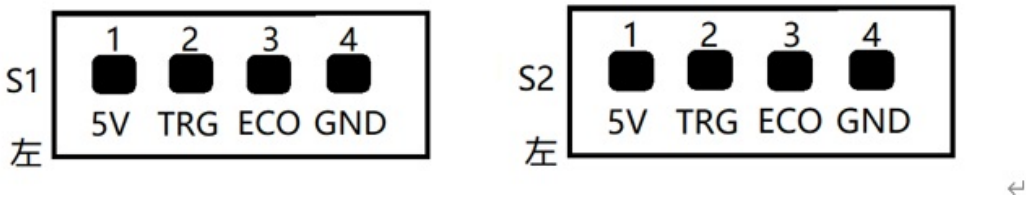
## Infrared obstacle avoidance module interface C1, C4, C2, C3



interface	description
5V	Module power supply positive
GND	Module power ground
ECHO	module output pin

Please note that the pin arrangement directions of C3 and C1, C4, and C2 are different. Usually, C1, C4, and C2 are used to detect obstacles in front of the chassis, and C3 is used to detect obstacles behind the chassis. Therefore, the arrangement direction is designed as different. When the module is triggered by an obstacle, the module ECO pin outputs a low level (driver feedback 0), and in other cases the module ECO outputs a high level (driver feedback 1).

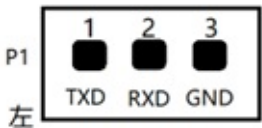
Ultrasonic ranging module interface S1, S2



interface	description
1. 5V	Module power supply positive
2. TRG	Module measurement trigger pin
3. ECO	Module measurement data output pin
4. GND	Module power ground

This driver supports the serial integrated ultrasonic ranging module, the measurement frequency of the driver is 5hz, and the unit of uploading data is meters.

Serial communication interface P1



interface	description
1. TXD	Drive serial data upload port
2. RXD	Drive serial command receiving port
3. GND	Serial ground

The communication connection mode of the driver is RS232 serial port, the serial port baud rate is 115200, 8 data bits, 1 stop bit, no parity check.

## communication protocol

The serial port baud rate is 115200, 8 data bits, 1 stop bit, no parity.

### computer commands

The direction motion commands are all 6-byte unsigned byte arrays. When the driver is installed face down, Ma1 corresponds to the right wheel motor, and Mb1 corresponds to the left wheel motor; when the driver is installed face up, Ma1 corresponds to the left wheel motor, and Mb1 corresponds to the right wheel motor.

#### Forward

0xcd	0xeb	0xd7	0x02	0x66	0xXX
Header	Header	Header	Length	Forward Command	Speed value, the value range is 0 to 100

#### Backward

0xcd	0xeb	0xd7	0x02	0x62	0xXX
Header	Header	Header	Length	Backward Command	Speed value, the value range is 0 to 100

#### Turn left

0xcd	0xeb	0xd7	0x02	0x63	0xXX
Header	Header	Header	Length	True Left Command	Speed value, the value range is 0 to 100

#### Turn right

0xcd	0xeb	0xd7	0x02	0x64	0xXX
Header	Header	Header	Length	Turn Right Command	Speed value, the value range is 0 to 100

#### Stop

0xcd	0xeb	0xd7	0x02	0x73	0xXX
Header	Header	Header	Length	Stop Command	The amount of braking, the value range is 0 to 100

A single motor control command is 13 bytes

0xcd	0xeb	0xd7	0x09	0x74	0xXX	0xXX	0xXX	0xXX	0xXX
Header	Header	Header	Length	Type	Right Motor Command	Left Motor Command	Reserved	Reserved	Value

The motor command has three states, 'F': forward, 'B': backward, 'S': brake The control amount range is 0 to 100,



E.g: tSSSS0000 is translated into hex as cd eb d7 09 74 53 53 53 53 00 00 00 00 means that all four motors are braked at 0% braking capacity Currently only the first two motors are valid, the first motor corresponds to the right wheel, and the second motor corresponds to the left wheel

## Drive upload packets

The drive uploads data at a frequency of 50HZ, and the format of the uploaded data package is: Header+length+content

Header: 3 u8 characters: 205 235 215

Length: 1 u8 character, the length does not include the Header and the length itself, the current packet length is 125, this length including end-of-string 0x00

Content: Consists of 25 numbers in 4-byte little-endian binary representation, separated by spaces 0x20.

The content of the complete data package constitutes a C language structure, and the specific structure of the structure is as follows:

```
typedef struct {
    int status;//Drive status, 0 means uninitialized, 1 means normal, -1 means error
    float power;//Supply voltage[12-46]v
    float theta;//angle [0-360]°
    unsigned int encoder_ppr;//The number of encoders corresponding to 1 wheel revolution
    int encoder_delta_r;//Right wheel encoder increment
    int encoder_delta_l;//Left wheel encoder increment
    int encoder_delta_car;//Center displacement of two wheels, unit
    unsigned int upward;//0 means face down installation, 1 means face up installation
    float max_speed;//Maximum speed, revolutions per second
    int hbz1;//The first infrared obstacle avoidance module status, 1 means trigger, 0 means no trigger
    int hbz2;//The second infrared obstacle avoidance module status, 1 means trigger, 0 means no trigger
    int hbz3;//The third infrared obstacle avoidance module status, 1 means trigger, 0 means no trigger
    int hbz4;//The forth infrared obstacle avoidance module status, 1 means trigger, 0 means no trigger
    float distance1;//The distance value of the first ultrasound module in cm
    float distance2;//The distance value of the second ultrasound module in cm
    float IMU[9];//mpu9250 9-axis data
    unsigned int time_stamp;//timestamp
}UPLOAD_STATUS;
```

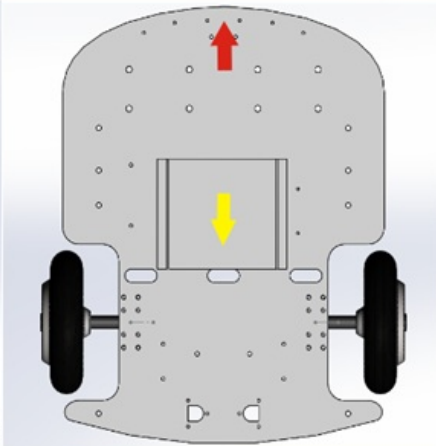
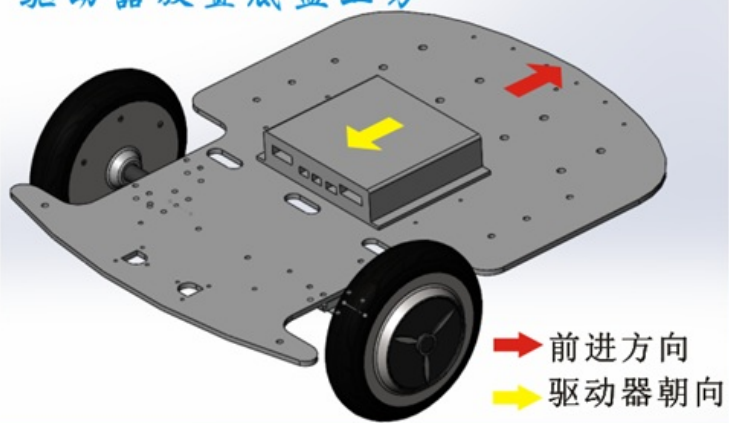
The data in the IMU[9] array is as follows

```
0: acceleration x
1: acceleration y
2: acceleration z
3: Angular velocity x
4: Angular velocity y
5: Angular velocity z
6: magnetometer x
7: magnetometer y
8: magnetometer z
```

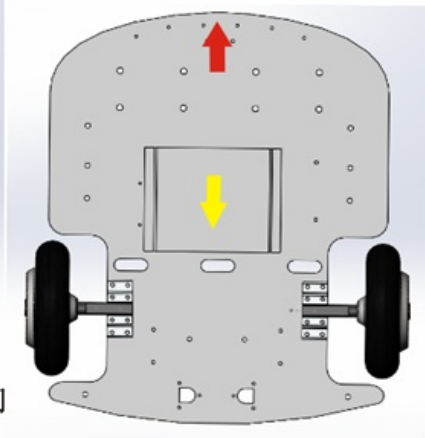
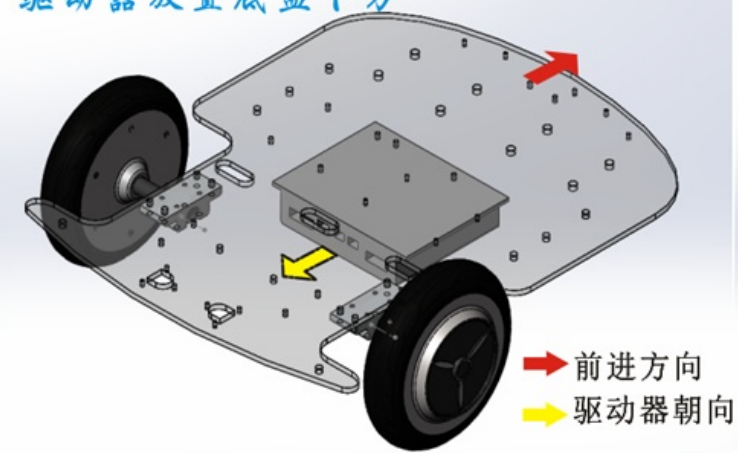
Please refer to the usage of the data package [ROS driver package](#)The corresponding part of the lungu branch code.

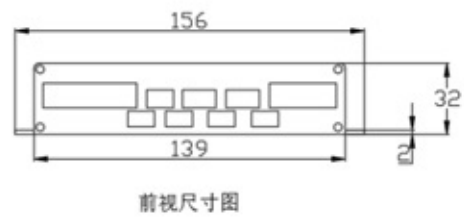
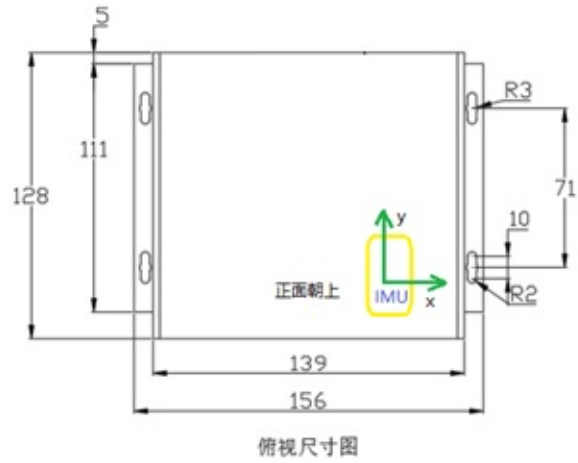
## Installation size

### 驱动器放置底盘上方



### 驱动器放置底盘下方

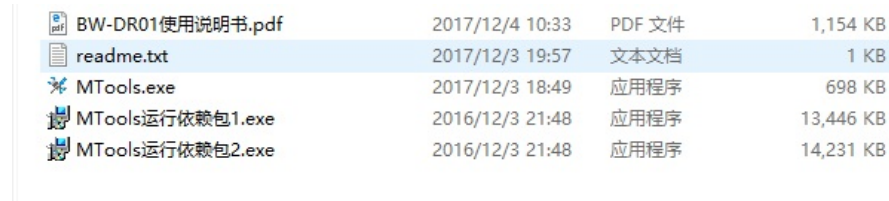









- [Software Installation](#)
  - [Serial driver parameter configuration](#)

## Software Installation

Open the driver package as shown below



 BW-DR01使用说明书.pdf	2017/12/4 10:33	PDF 文件	1,154 KB
 readme.txt	2017/12/3 19:57	文本文档	1 KB
 MTools.exe	2017/12/3 18:49	应用程序	698 KB
 MTools运行依赖包1.exe	2016/12/3 21:48	应用程序	13,446 KB
 MTools运行依赖包2.exe	2016/12/3 21:48	应用程序	14,231 KB

First double-click to install MTools running dependency package dep1.exe and dep2.exe.

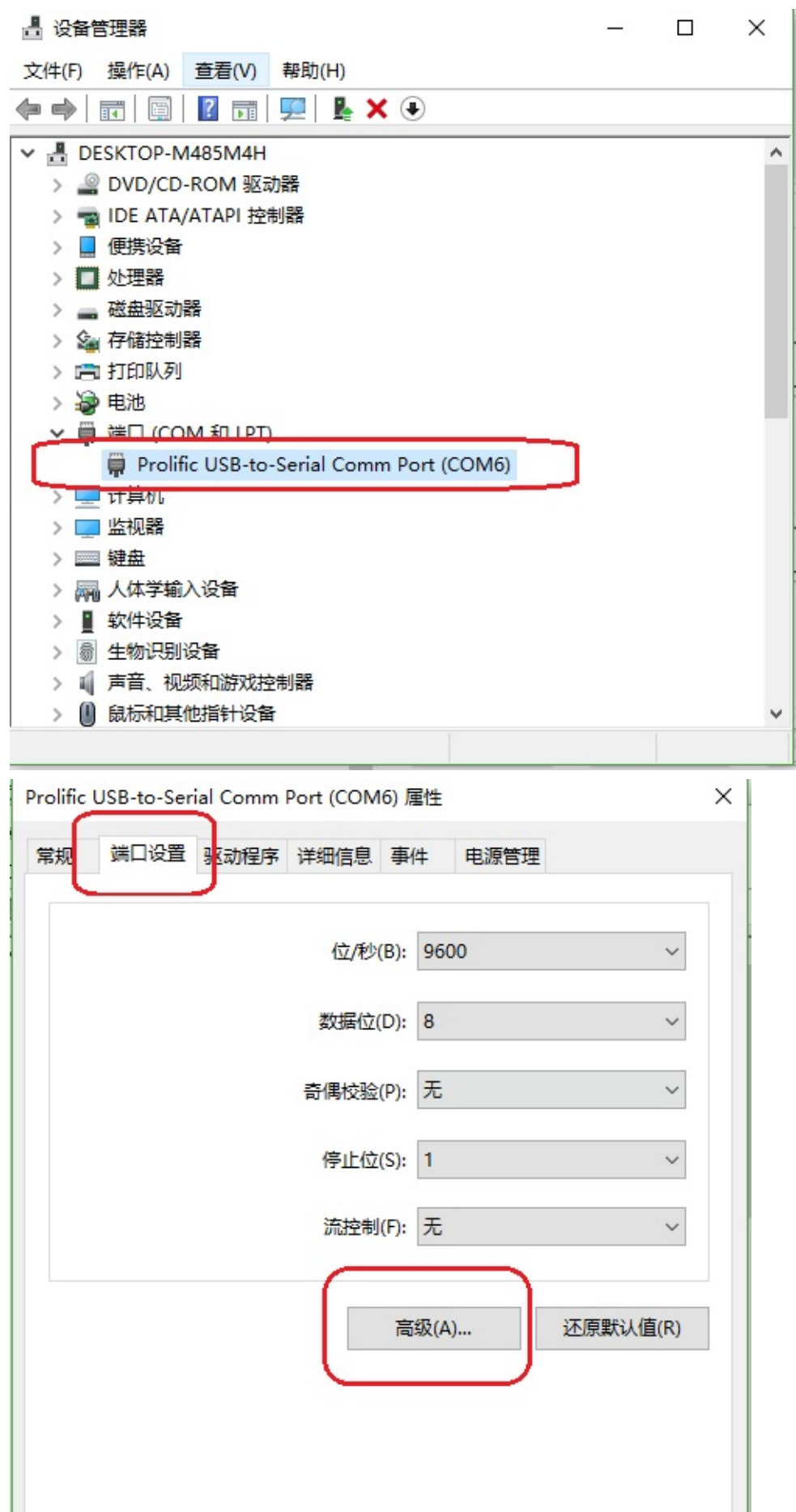
MTools.exe is the drive debugging software, double-click to run directly. It is recommended to create a software shortcut on the desktop to facilitate subsequent use. Shortcut creation steps: MTools.exe right-click -> "Send to" -> "Desktop Shortcut"

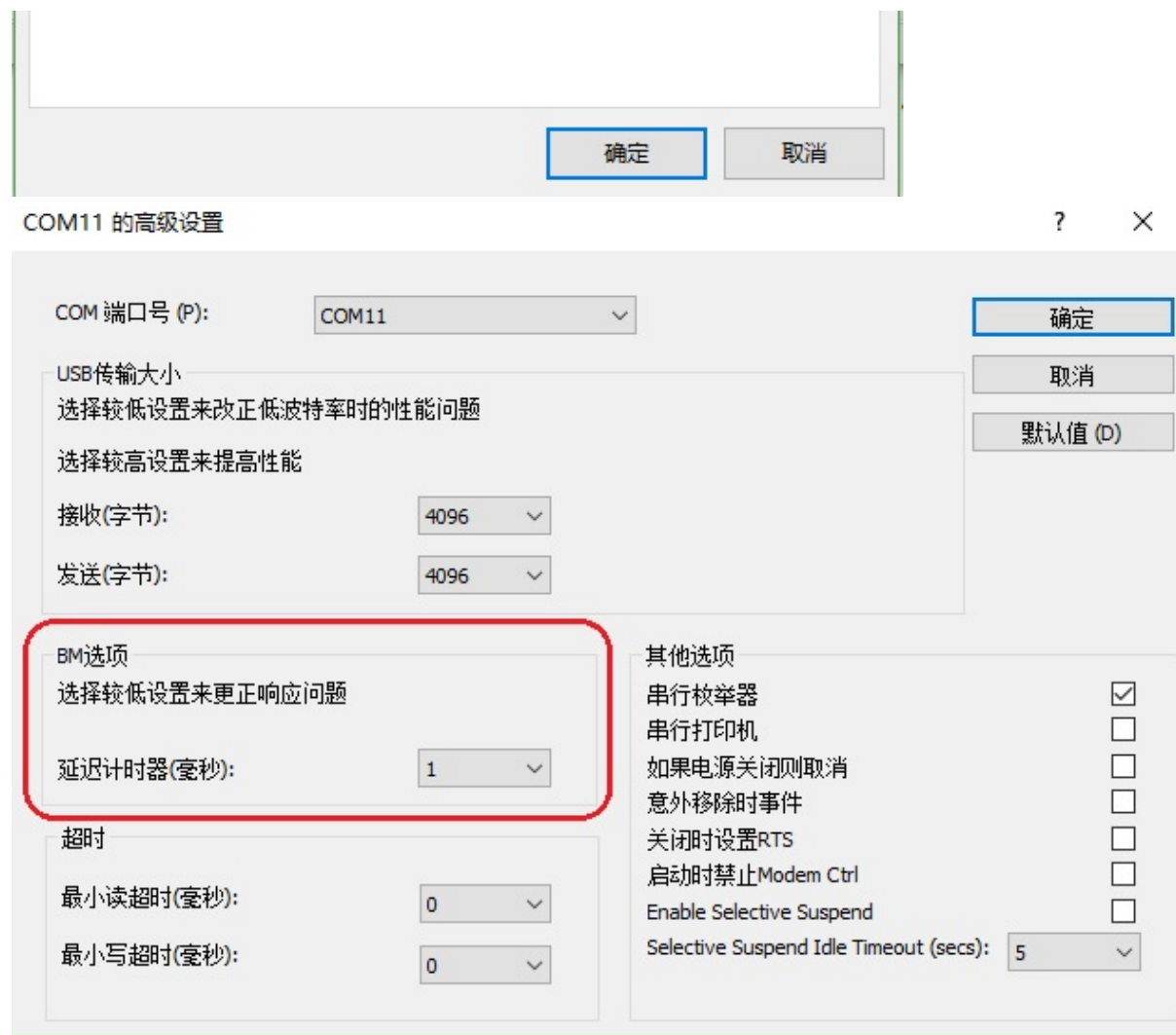
## Serial driver parameter configuration

The communication connection between MTools.exe software and the driver is rs232 serial port, the serial port baud rate is 115200, 8 data bits, 1 stop bit, and no parity check.

For windows system, the driver of the usb-to-serial module has a delay parameter. If the default value is unreasonable, the uploaded data cannot be processed in time by the driver. Please set it to the minimum value according to the following figure (if this option does not exist, do not set it).







- Overview
  - software interface
  - 2.2 Precautions

## Overview

Through the MTools.exe software, the basic operating parameters of the drive can be configured, and the motor can be easily debugged and tested online. This software can preview the data uploaded by the IMU and sensor peripherals in real time, and can assist in diagnosing the health status of drivers and peripheral modules.

## software interface



The software interface is mainly divided into three areas: serial port configuration panel, data preview panel, and control panel. The size of the interface can be adjusted arbitrarily. The blue status bar at the bottom of the interface will provide real-time feedback information according to the operation. Please pay attention to the actual prompt content.

After opening the serial port and connecting the drive, the data preview panel will update the data in real time.





After the driver is powered on, there will be a period of initialization time, and the status will be 0. After the initialization is completed, the status will become 1, and the background color will become green. Only in this case can the "Control Panel" be operated.



## 2.2 Precautions

A. When closing the software or closing the serial port, if the software crashes, please use the "Task Manager" to force close the software, and then re-plug the USB serial port device. After restarting the software, it will return to normal.

b. When the driver is connected to the motor for the first time for debugging, if the Hall line sequence is not determined, please operate Chapters 4 and 5 first.

Otherwise, misoperation will easily damage the driver switch tube.

- [Configure drive parameters](#)
  - [poles number](#)
  - [Maximum speed control](#)
  - [The installation direction corresponds to the left and right motor channels](#)
  - [Generic actions](#)

## Configure drive parameters

There is an initialization time after the drive powers up with a status of 0. After initialization is complete, the status changes to 1 and the background color changes to green, in which case the Control Panel can be operated.

驱动器参数设置

电机极对数:	15	读取	设置
最大转速(RPM):	300	读取	设置
正向安装:	False	读取	设置

### poles number

The number of motor poles means the number of magnetic pole pairs, and the usual hub brushless motor value is 15 or 12. Click "Read" to get the current setting value of the drive, click "Write" to send the current value to the drive ROM. If the actual number of the magnetic pole pairs of the motor is unknown, it can be calculated by counting the encoder count value recorded by the software when the motor rotates once.

电机编码器计数(个)

左轮电机(通道2):	0	右轮电机(通道1):	0
------------	---	------------	---

Number of pole pairs = 1 turn count ÷ 6

### Maximum speed control

Click "Read" to get the current setting value of the drive, click "Write" to send the current value to the drive ROM. The maximum speed of the motor is recommended to be set to the value actually needed, rather than the maximum value that the motor can achieve, which can improve the accuracy of low-speed control. For the maximum speed that the motor can achieve, please issue a speed control signal of 100 in the "speed open loop" mode, and the speed feedback value is the maximum speed, and the maximum achievable speed is related to the battery level.

电机转速(RPM)

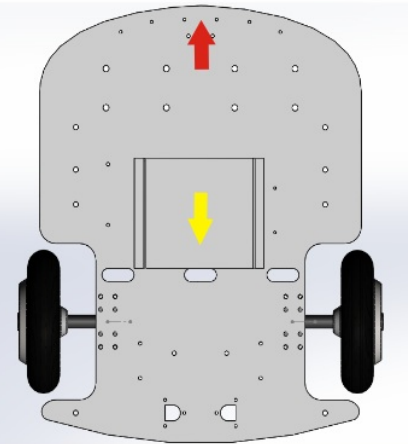
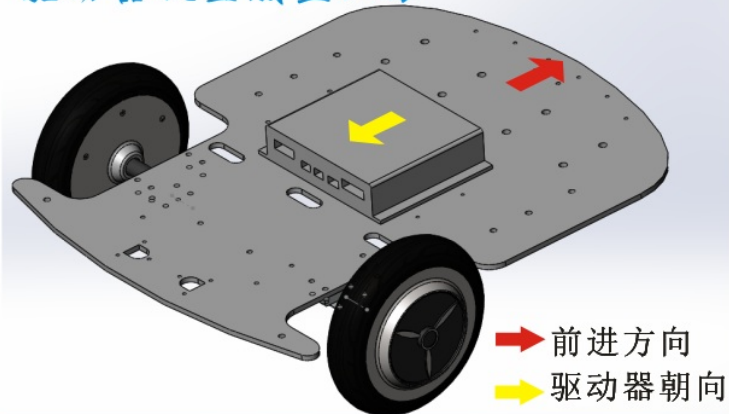
左轮电机(通道2):	0	右轮电机(通道1):	0
------------	---	------------	---

### The installation direction corresponds to the left and right motor channels

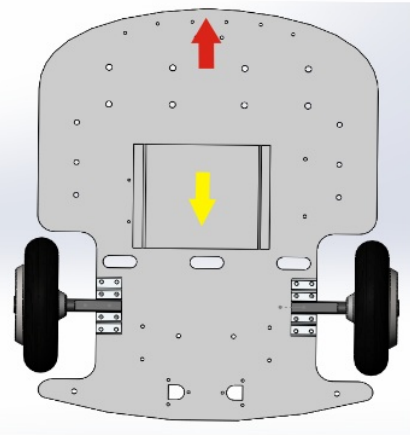
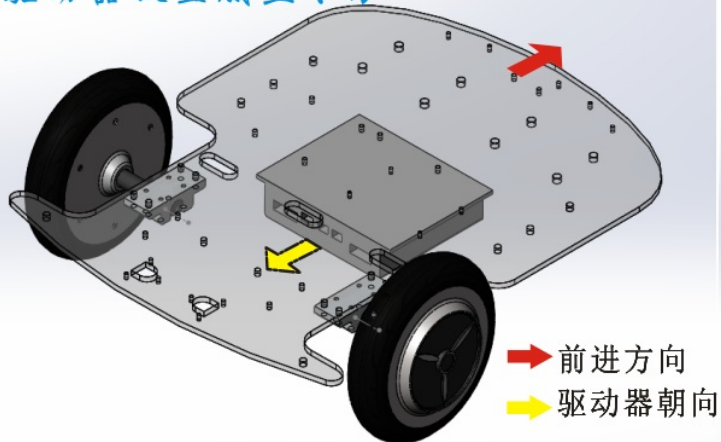
Click "Read" to get the current setting value of the drive, click "Write" to send the current value to the drive ROM, the value can only be false, true these two cases. Forward mounting refers to the horizontal mounting of the drive face up, and reverse mounting refers to the horizontal mounting of the drive face down. In both mounting directions, in order to facilitate wiring, the corresponding wiring ports of the motor on the left and right of the drive are opposite. We assume that the drive is a rear-drive motor.

Motor	Forward (True)	Reverse (False)
Left Motor	Ma1 H0	Mb1 H1
Right Motor	Mb1 H1	Ma1 H0

驱动器放置底盘上方



驱动器放置底盘下方



## Generic actions



"Restart" makes the drive control chip restart, "e-stop" can brake two motors at the same time, "flash" configures the drive to be in the firmware update state, "open loop" temporarily configures the drive to speed open-loop control mode, and "calibration" starts the drive IMU automatic calibration program.



- [Test the motor phase and Hall wire wiring sequence](#)
  - [Velocity Open Loop Mode](#)
  - [Determining the Hall Line Sequence](#)
  - [closed-loop speed control test](#)
  - [Adjust the forward rotation direction of the motor](#)

## Test the motor phase and Hall wire wiring sequence

For this drive, theoretically any motor phase wire wiring, there is a set of corresponding Hall wiring to make the motor operate normally, so you can first arbitrarily specify the motor phase wiring sequence, here we specify mA to connect the motor yellow wire, mB to connect the motor green line, mC to connect the motor blue line; The following describes how to determine the Hall line order.

When the Hall line sequence is incorrect, the driver needs to be set to "speed open-loop mode", and the value of the speed should not be greater than 20, which can prevent abnormally high current from damaging the motor coil and drive tube. The line sequence of the two motors is the same, that is, the motor lines and Hall lines of the left wheel and the right wheel are in exactly the same order, so the following test is only for the left wheel motor.

### Velocity Open Loop Mode



Click "open loop", the drive will temporarily enter the "speed open loop" control mode, at this time, the motor PWM switch value is directly proportional to the speed command. Restarting the drive will cause the drive to exit the "speed open loop" control mode and enter the default speed closed loop mode.

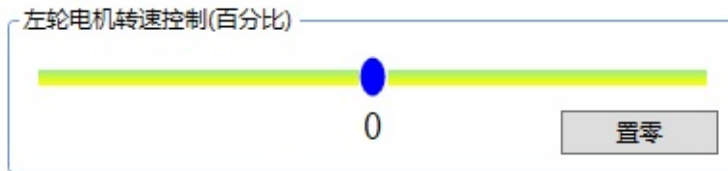
### Determining the Hall Line Sequence

Make sure that the driver is in the "speed open loop" mode. Taking the left motor as an example, it is assumed that the colors of the three Hall wires of the motor are yellow 1, green 2, and blue 3 respectively.

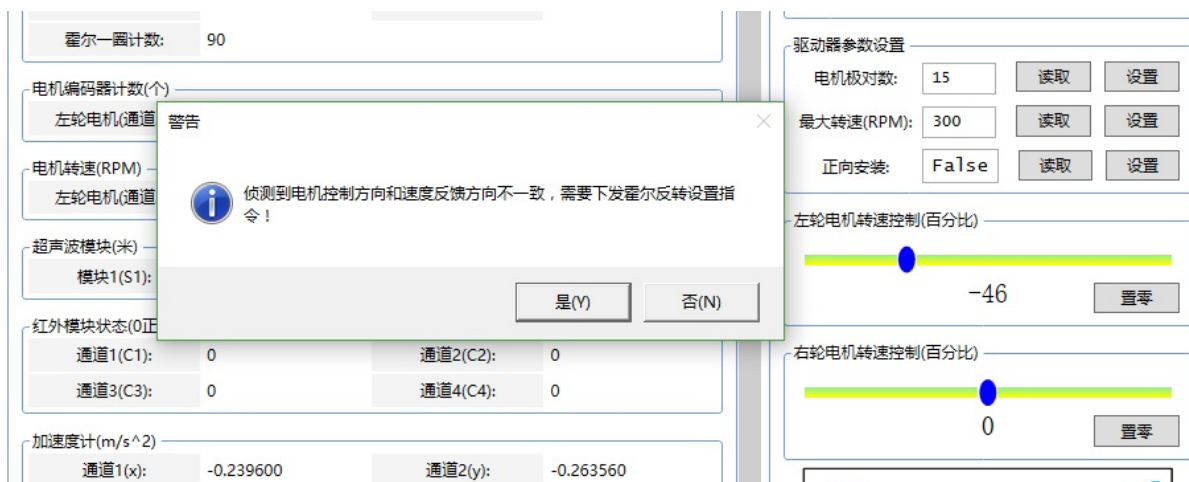
Hall wire interface	order 1	order 2	order 3	order 4	order 5	order 6
HA	Y1	Y1	G2	G2	B3	B3
HB	G2	B3	Y1	B3	Y1	G2
HC	B3	G2	B3	Y1	G2	Y1

Try the 6 wiring sequences in the above table in turn, for each wiring sequence:

- Set speed 10% to see if the motor runs smoothly. If it is not normal, the sequence of this group is wrong, please "zero" and then test the next group of wiring; if it is normal, perform operation b.
- Set speed -10% to see if the motor runs smoothly. If it is not normal, it means that the sequence of this group is wrong, please "zero" and then go back to a to test the next group of wiring; if it is normal, it means that the Hall line sequence is correct , continue to step c.

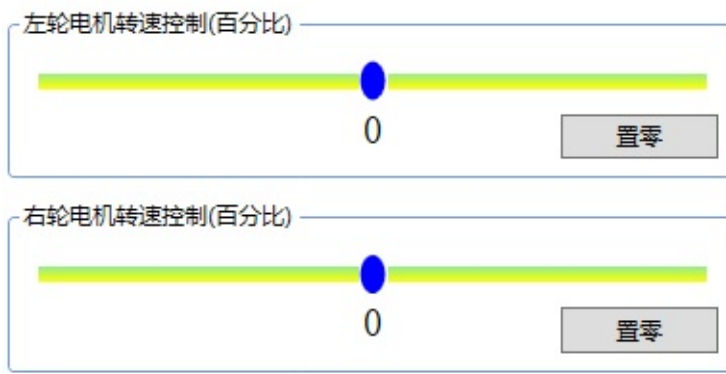


c. Now the motor can basically run normally. Finally, it is necessary to make sure that the direction of the control speed and the direction of the speed feedback are consistent. For example, if the control speed greater than zero is sent, the speed should normally return to a speed greater than zero, so that the speed can be closed-loop. First increase the speed control amount in the forward direction, so that the speed is greater than 200RPM or less than -200RPM; then increase the speed control amount in the reverse direction, and also make the speed greater than 200RPM or less than -200RPM. After the operation is completed, the software will automatically determine whether the control direction is opposite to the direction of the speed. If the prompt shown in the figure below appears, press "Yes". If it does not appear, it means that the debugging has been completed and the speed closed-loop test can be continued.



## closed-loop speed control test

According to the Hall line sequence determined in the previous section, connect the right wheel motor to the driver, then restart the driver, and wait for the driver to be initialized. Now arbitrarily adjust the slider in the figure below to set the motor speed.



While controlling, you can also observe the speed feedback value.



## **Adjust the forward rotation direction of the motor**

On the premise that the speed closed-loop control is successful, if it is found that the forward rotation direction of the motor is opposite to the direction you need, first judge whether the installation direction is set correctly according to Section 3.3. If the installation direction is set correctly, then adjust the ma and Mb of the motor phase line. , At the same time, the HA and HC of the Hall line are reversed, so that the forward rotation direction of the motor can be reversed.



- [Drive firmware update](#)
  - [Set the drive to flash mode](#)
  - [Update the drive firmware using the Flash Loader Demo tool](#)

## Drive firmware update

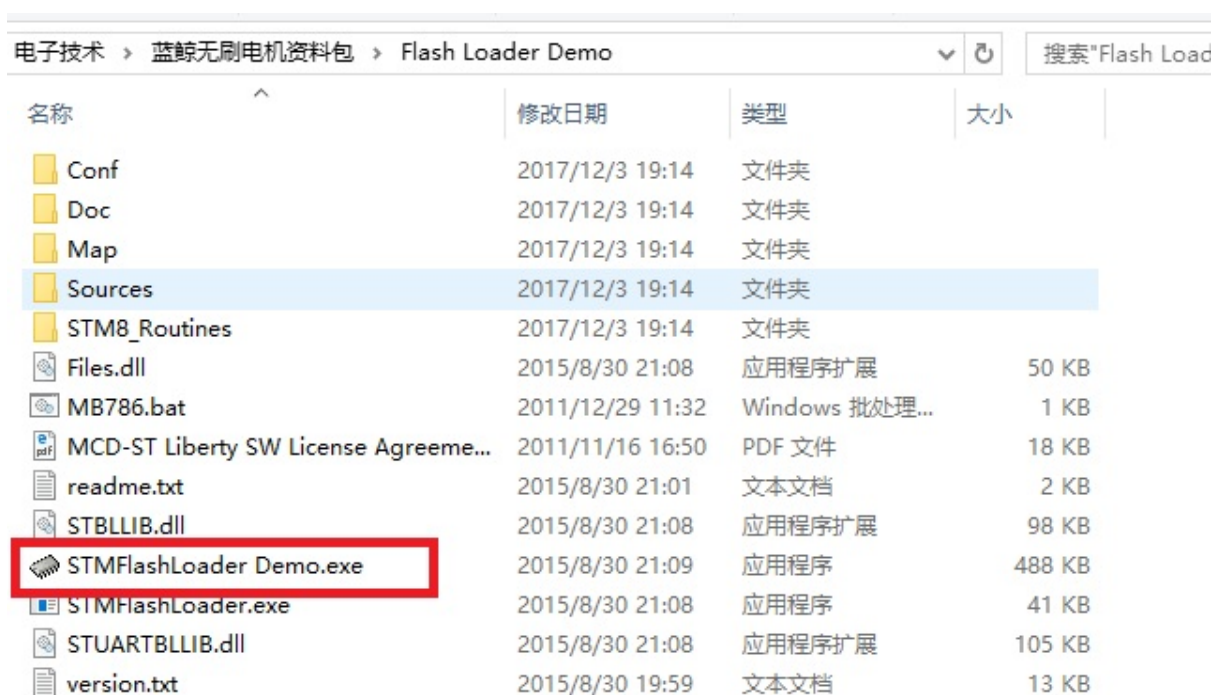
### Set the drive to flash mode



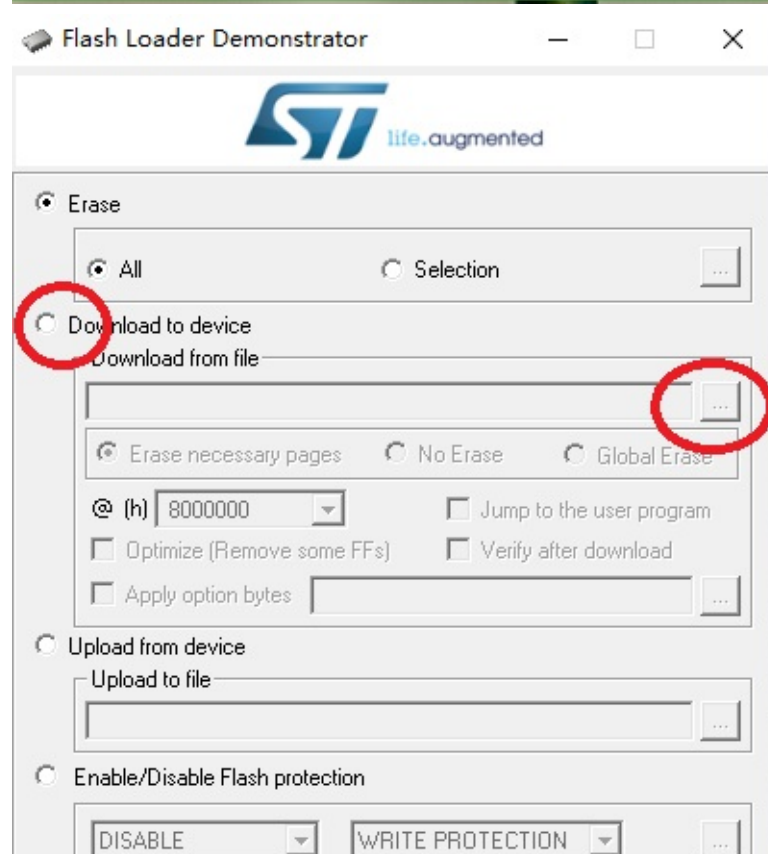
After opening the serial port, click "Flash" to set the drive to flash mode. After entering the flash mode, please close the drive software and continue to the next step.

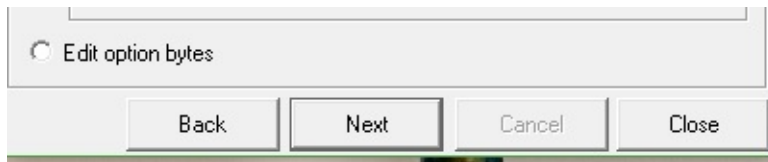
### Update the drive firmware using the Flash Loader Demo tool

Open STMFlashLoader Demo.exe

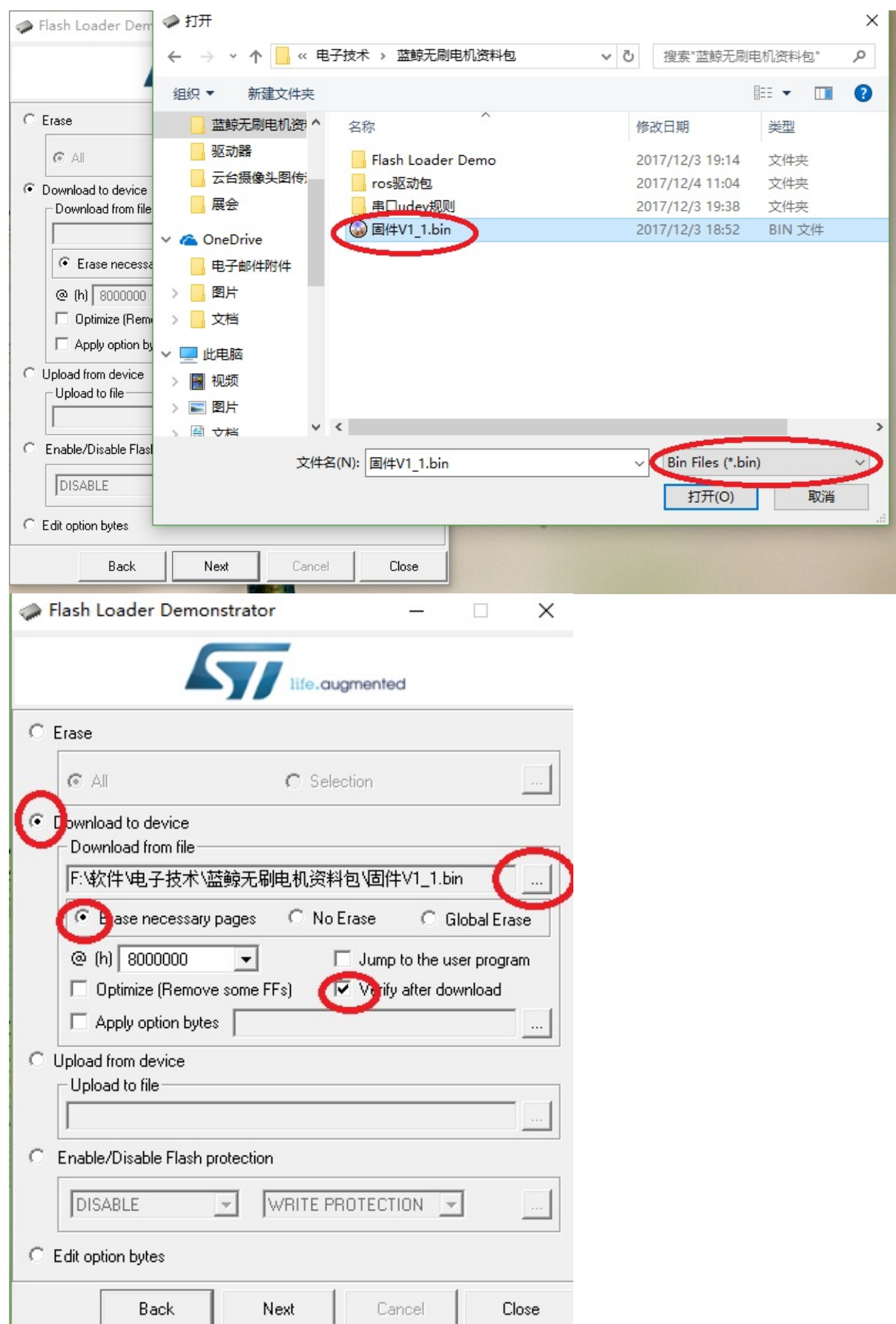


Select the serial port number, and keep clicking "next" until the burning option interface appears.

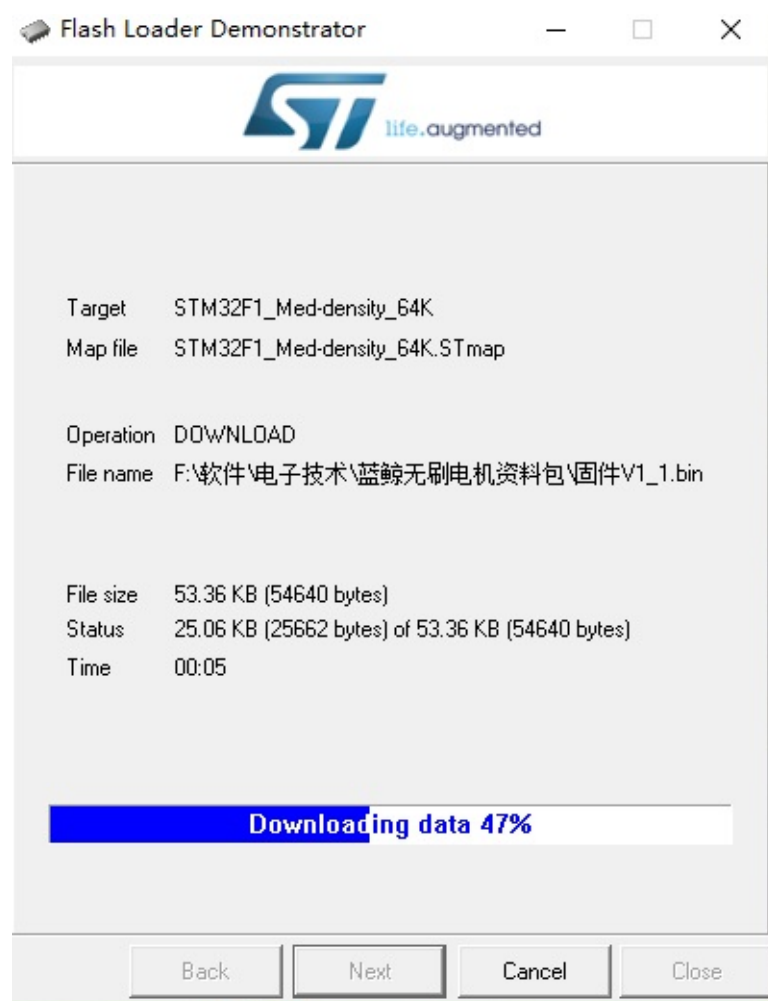




Select the second "Download to device" option to load the firmware file.



Click "next" to start the firmware update.



After the firmware is updated successfully, close the software, power on the drive again, and it can be used normally.



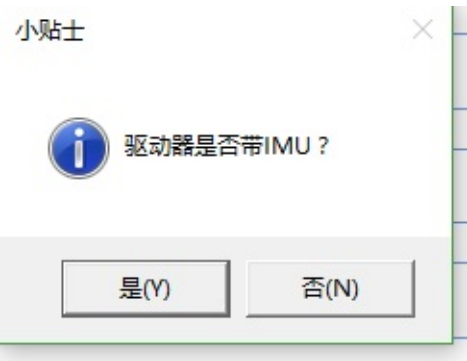
- [Calibrate IMU](#)

## Calibrate IMU

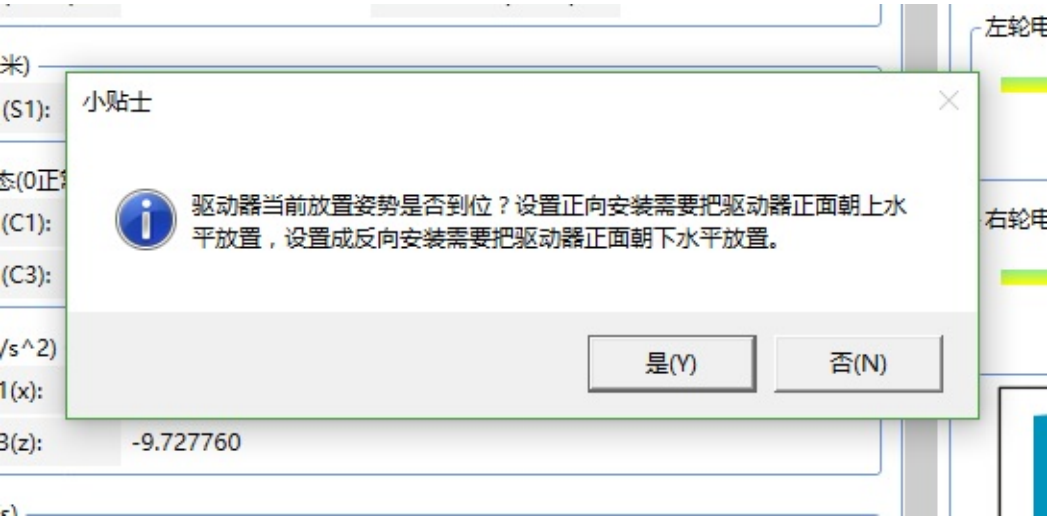
The calibration process can only be performed if the drive has an IMU. First, place the drive horizontally according to the "forward installation" setting. Then, click "Calibrate" to put the drive into the IMU calibration routine.



Select "Yes"



Select "Yes" to start calibration



After calibration starts, the gyroscope data will become "0"

电机转速(RPM)

左轮电机(通道2): 0 右轮电机(通道1): 0

超声波模块(米)

模块1(S1): 小贴士

红外模块状态(0正常)

通道1(C1): 驱动器已经进入IMU校准模式，校准过程中保持驱动器静止，等待两分钟，陀螺仪有非零数据时表示校准完成！

通道3(C3):

加速度计( $m/s^2$ )

通道1(x):

通道3(z): 0.000000

陀螺仪(rad/s)

通道1(x): 0.000000 通道2(y): 0.000000

通道3(z): 0.000000

磁场(uT)

通道1(x): 0.000000 通道2(y): 0.000000

通道3(z): 0.000000

Wait two minutes, after the calibration is over, the gyroscope data will appear non-zero. After the calibration is successful, the data value of channel 3 is generally less than 0.004.

陀螺仪(rad/s)

通道1(x): -0.001068 通道2(y): 0.001068

通道3(z): 0.001068



- [ROS Driver Instructions](#)
  - [Overview](#)
  - [Hardware connection](#)
  - [Ubuntu Host serial port permission setting and device number mapping](#)
  - [package compilation](#)
  - [executable nodes](#)
    - [xqserial\\_server Node](#)
    - [Subscribed topic data](#)
    - [Published topic data, 50hz frequency](#)
    - [Node parameters](#)
    - [Published tf transform](#)
    - [Calibration IMU](#)
    - [Calibrate Robot Parameters](#)

## ROS Driver Instructions

### Overview

The ROS chassis driver `xqserial_server` package is responsible for the differential solution of the subscribed `/cmd_vel` topic, and then converts it into a serial port command for the motor driver. At the same time, the serial port package information uploaded by the driver is processed, and then organized and published into the topic of odometer and sensor data.

### Hardware connection

The motor driver and the host are connected through a USB to rs232 module (corresponding to "motor serial port"). The serial port should be bound to the USB port through udev rules and mapped to `ttyUSB001`. Serial port parameters are baud rate 115200, 8 data bits, 1 stop bit, no parity.

Please operate the wiring of the driver according to the driver manual. Before debugging the `xqserial_server` package, you need to ensure that the `mtools` software can normally control the forward and reverse rotation of the motor in a closed loop, otherwise the driver may be burned.

### Ubuntu Host serial port permission setting and device number mapping

We are using a usb to rs232 module, and ubuntu will recognize it as a `ttyUSB0` device by default. This can be queried with the `ls /dev` command. The identified device number is random, in order to map the serial port number to `ttyUSB001`, and set the device user read and write permissions at the same time. We can do this with udev rules.

For details, refer to Step 3 of the following post. [The use and realization principle of automatic charging function package](#)

### package compilation

It involves the basic knowledge of ros workspace, ros package, `catkin_make`, ros workspace source command, and git. If you are not sure about these, please learn them from Google.

First git clone to download the `xqserial_server` package and put it in the `src` folder of the ros workspace, switch to the `lungu` branch, and finally compile it with `catkin_make`.

Assuming your ros workspace is in ~/catkin\_ws, the following are all instructions for downloading and compiling.

```
cd ~/catkin_ws/src/
git clone https://github.com/BluewhaleRobot/xqserial_server.git
cd xqserial_server
git checkout lungu
cd ~/catkin_ws/
catkin_make
```

## executable nodes

After catkin\_make is compiled, an executable node xqserial\_server will be generated, and a launch file xqserial.launch that can be used directly has been provided in the launch folder. visualization.py in the script folder is a python script file that can be used to visualize the IMU data published by xqserial\_server.

## xqserial\_server Node

The xqserial\_server node is the master node of the package, which involves the processing and publishing of data such as differential speed calculation, odometer, sensor, and tf. In hardware, this node accesses the serial device of the driver. The ROS configuration information for this node will be described in detail below.

## Subscribed topic data

Topic	Message type	description
/cmd_vel	geometry_msgs/Twist	The topic of speed is resolved into motor control commands after subscription.
/imu_cal	std_msgs/Bool	IMU auto-calibration command, set to true to trigger auto-calibration. At this time, the robot needs to wait for 2 minutes, and cannot be triggered repeatedly within 2 minutes.
/globalMoveFlag	std_msgs/Bool	When set to true, the /cmd_vel command takes effect; when set to false, /cmd_vel is disabled.
/barDetectFlag	std_msgs/Bool	Set to true to enable ultrasonic obstacle avoidance, false to disable ultrasonic and infrared obstacle avoidance.

## Published topic data, 50hz frequency

Topic	Message type	description
/xqserial_server/Odom	nav_msgs/Odometry	Chassis odometer, angle part has been integrated with IMU.
/xqserial_server/StatusFlag	std_msgs/Int32	A value of 0 indicates that the driver is still initializing, and a value of 1 indicates that processing state is normal. A value of 2 means IR is triggered.
/xqserial_server/Twist	geometry_msgs/Twist	The speed of the chassis feedback, the linear speed is the encoder feedback value, and the angular speed is the IMU return value.
/xqserial_server/Power	std_msgs/Float64	Battery voltage in V.
/xqserial_server/Pose2D	geometry_msgs/Pose2D	Two-dimensional pose data, you can easily view the coordinates and angles of the odometer.
/xqserial_server/IMU	sensor_msgs/Imu	IMU topic data, the tf relationship of IMU needs to be adjusted in launch according to the following settings.

/xqserial_server/Sonar1	sensor_msgs/Range	Ultrasonic module S1 topic data, tf relationship needs to be adjusted in launch according to the following settings.
/xqserial_server/Sonar2	sensor_msgs/Range	Ultrasonic module S2 topic data, tf relationship needs to be adjusted in launch according to the following settings.

## Node parameters

Parameter	Type	Defaults	Description
~port	String	/dev/ttyUSB001	Motor driver serial port name
~wheel_separation	double	0.36	The distance between the centers of the power wheels or tracks on both sides, in meters
~wheel_radius	double	0.0825	The radius of the power wheel or track, in meters
~max_speed	double	5.0	The maximum speed of the wheel or the maximum speed of the crawler power wheel, in revolutions per second

## Published tf transform

odom → base\_footprint 50hz Publishing frequency, base\_footprint in the center of the car.

## Calibration IMU

Place the trolley horizontally and still, and publish the ros topic to start the IMU self-calibration. During the calibration process of about 2 minutes, the trolley cannot be moved or collided.

```
rostopic pub /imu_cal std_msgs/Bool '{data: true}' -1
```

During the calibration process, the /xqserial\_server/IMU topic data is always zero, and returns to the normal value after the calibration is completed. Through this topic data, the calibration progress and calibration result can be judged.

## Calibrate Robot Parameters

Before performing the following operations, you need to complete the IMU calibration in step 6, because the IMU will affect the odometer feedback accuracy.

## Calibrate wheel\_radius

Calibration principle:

The tape measure can measure the actual distance of the car (set to L), and the topic /xqserial\_server/Pose2D can check the feedback distance of the car odometer (set to l), so we can get the ratio of L/l (the value after calibration should be 1), this ratio is the scaling factor of the wheel radius parameter wheel\_radius in the launch file. Multiply this ratio by the current value to get the calibrated value.

```
After calibration wheel_radius = Before calibration wheel_radius * L / l
```

Calibration steps:

Remote control the car to the starting line, the angle of the car should be aligned, and then restart the xqserial\_server package launch file to reset the odometer to zero. Then remote control car moves about 2 meters in a straight line, record the forward distance L of the car measured with a tape measure, and the topic feedback value l, and calculate the L/l ratio. Obtain the calibrated wheel\_radius value according to the calibration principle, modify the wheel radius parameter in the launch file to this new value, and then restart the launch file, the radius calibration is completed.

## Calibrate wheel\_separation

Note: The calibration of the wheel radius needs to be completed before calibration.

Calibration principle:

The angular velocity feedback value in the /xqserial\_server/Twist topic is calculated by the IMU, so it can be used as a reference value  $W$ , and the angular velocity value in the /cmd\_vel topic is the set value  $w$ . The calculated  $W/w$  ratio is the scaling factor for the wheel spacing parameter. This value does not need to be very precise, because it will only affect the accuracy of the angular velocity control, not Affects the accuracy of the angle part of the odometer. If you want to get accurate values, it is recommended to write a python node to subscribe to these two topics, and get a more accurate ratio by calculating the average value.

```
After calibration wheel_separation = Before calibration wheel_separation * W / w
```

Calibration steps:

remote control car rotates in place, and the calibrated wheel\_separation value is obtained according to the calibration principle. Modify the wheel spacing parameter in the launch file to this new value, then restart the launch file, and the wheel spacing parameter calibration is complete.